

# UM::Autonomy Thurman

Cyrus Anderson, Noah Ehmann, Anthony Huang, Thomas Huang, Jeremy Lipshaw, Kunjan Singh, Dan Snyder, Rebecca Sorgenfrei, Minh Tran, Anthony Uytingco, Marisa Witcpalek

**Abstract** – UM::A Thurman is a fully autonomous surface vehicle with a custom hull designed for maximum maneuverability and stability while maintaining small principle dimensions. This boat was designed to compete in the tenth annual AUVSI Foundation RoboBoat competition. As a part of this competition Thurman will navigate an obstacle course, dock at specified sign, identify acoustic beacons, and launch an autonomous submarine to locate an underwater screen. Our main focuses this year were integrating the electrical system into the hull and improving our SLAM algorithm. This paper details the changes we made to meet these goals as well as the improvements made on previous designs based on lessons learned.



Fig. 1. UM::Autonomy's 2016 boat, Thurman.

## I. INTRODUCTION

Thurman is UM::Autonomy's 2016 entry for the AUVSI RoboBoat competition. Like our previous entries, Thurman is designed to improve on lessons learned from previous designs as well as complete all challenges set forth by the competition. As it would be impractical to describe the design in full, this paper focuses on the improvements made over previous designs.

A few of the notable design changes include the transition to a monohull design instead of catamarans, the integration of the electrical system into the hull, and the re-writing our SLAM implementation. The details of these changes will be enumerated in the following sections.

## II. HULL AND DECK

We have radically changed the design of the hull and deck for this year's competition. This section will discuss our design decisions, construction process, hull attachments, and sensor mounts for the hull and deck.

### A. Hull Design

Our main goals this year were to have more room on our deck for the interoperability challenge vehicle, decrease weight, and allow easier buoy navigation. As such, our design process focused on stability, maneuverability, and small principal dimensions. In the end, we decided upon the professionally praised Seabright Skiff (nature of boats) which

greatly diverges from our previous designs as it consists of a monohull. The decision to transfer to a monohull was capitalized by the fact that this design provides a shallower draft, narrower beam, and a higher capacity for storage.

Like Thurman's past iteration, SharkByte, Thurman is very close to the water for increased stability and control. Unlike SharkByte, however, Thurman has a large, hollow interior which we effectively used as our electrical box which will be discussed more thoroughly in Section III, A. Thurman also has a flat bottom so it can easily balance on tables or other flat surfaces for easier transportation.

### B. Hull and Deck Construction

Initially, the hull was digitally drawn using the computer-aided drafting (CAD) software, CATIA. Through the software, a toolpath was generated that could cut the shape of the hull from closed-cell insulation foam on a CNC routing machine. To accommodate the working height of the router, the model was cut into multiple cross sections parallel to the waterline and eventually glued together to create the representative foam product.

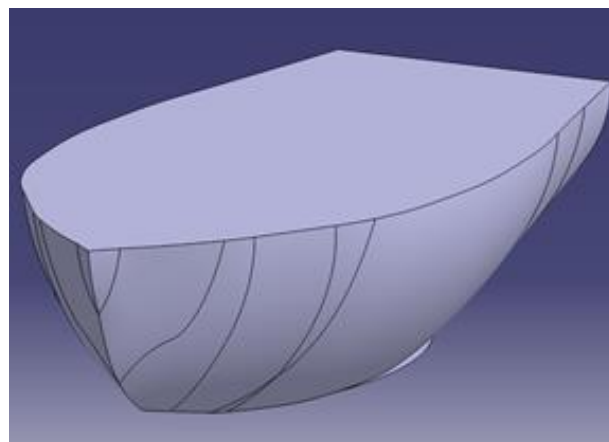


Fig. 2. The CAD drawing of the hull used to generate the toolpath and cut out the hullform.



Fig. 3. The finished hull after applying fiberglass to the hullform and removing the foam.

Next, woven 80-20 fiberglass (fiberglass with eighty percent of the fibers running in one direction and twenty percent running in the perpendicular direction) cloth was cut into large sections to efficiently cover the maximum amount of area of the foam. Epoxy resin and hardener was used to permanently place the fiberglass cloth and provide the increased stiffness and strength. Three layers of fiberglass were added to the foam to maintain structural integrity and waterproofness. Finally, to aid in transportability and to provide attachment points for the crane at competition, we attached four stainless steel handles via fiberglass symmetrically around the boat.

The deck is made of a thin plate of aluminum that has been water-jetted to fit the exact dimensions of the top of the boat. Veering away from the thick honey-comb material as commonly used in previous designs, this aluminum deck, while slightly heavier, is less expensive and easier to drill through. To retain the hull's waterproofness, we added attachments such as the outriggers to the deck and not to the hull. In this way, there is a much lower probability that water can get inside the holes from splash if they were drilled into the deck in comparison to if they were drilled directly into the hull.

As briefly mentioned, attached to the deck are a set of outriggers. These outriggers, which have an adjustable height, have two main purposes: to provide roll stabilization and to mount the thrusters. The outriggers are made of light-weight PVC piping, with a large, hollow two inch diameter pipe at the bottom of the system to add buoyancy. An additional PVC attachment was placed at the front of the boat used to attach the Velcro to the docks and hook up the hydrophone. The hydrophone implementation will be described in more detail in Section C.



Fig. 4. The hull with outriggers attached

### C. Sensor Mounts

This sub-section will present the types of sensors and the mounts used in this iteration of UM::Autonomy's autonomous boat.

We used the same general design for the waterproof camera mount as we have last year. The camera bottle consists of three main parts: the top case, the bottom case, and two lenses. The top case provides half of the camera mount, two frames for the lenses' integration, and a slit for the O-ring. Moreover, the bottom case provides the second half of the camera mount,

integration to the boat, and ridges to accommodate the O-ring and top case. Both of the cases were 3D printed in the University of Michigan's 3D lab. Respectively, the front and back Acrylic lenses, fabricated by a laser cutter, allow the camera to see clearly through the case and allow us to know if the camera is on via the light on the back of the camera. The front of each lens is covered in a polarized to film to reduce glare to the cameras. The bottom case is attached to the boat through the sensor tower and the top case slides directly into the bottom case. By bolting the cases together, the O-ring seal will be complete and the camera will be successfully waterproof. The camera is elevated as far above the surface of the water as the competition height limits allow to obtain the widest field of view.

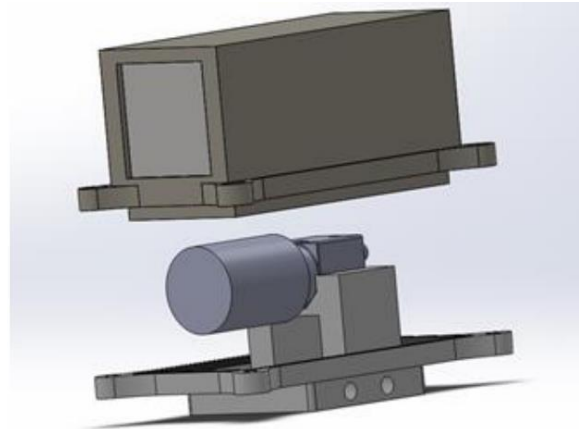


Fig. 5. A CAD model of the camera bottle used this year.

Thurman's LIDAR collects data from a single 2D plane. In order to create a 3D point cloud, the LIDAR is mounted on an AX12 servo. This custom rocking LIDAR mount was fabricated using a 3D printer. Like last year, the LIDAR is mounted on top of the deck to avoid splashes.

Thurman is equipped with a Garmin GPS. It is mounted directly on the deck beside the electrical box.

We are using an Aquarian Audio Products H2a high impedance hydrophone to detect the locations of the acoustic beacon. This hydrophone is normally omnidirectional, but we inserted it into a long section of PVC pipe to use it as a directional hydrophone. This housing is mounted below the PVC attachment used for Velcro placement found on the front of the boat.

### D. Interoperability Challenge

Due to time constraints and the late addition of this challenge we chose to purchase a pre-made RC sub to use for this challenge. We chose to use a Huiqi HQ2011 submarine, seen in Figure 6. This submarine is 11.00" x 8.65" x 6.29". The sub has built-in headlights and it will surface if it loses radio communication. We added our own camera on top because there is no built-in camera.

In order to easily launch and retrieve the sub, we chose to tether it to Thurman. When we want to launch the sub, we release some slack on our tether cable, causing the sub to fall into the water, in order to retrieve the sub we do the opposite and reel in the tether. The tether is wrapped around a large reel



and a motor with a custom gearbox rotates this reel to lower or raise the AUV.



Fig. 6. This is the RC submarine we purchased to modify and use for our AUV.

### III. ELECTRICAL SYSTEM

The following section will describe the changes we have made to the electrical system of Thurman compared to previous boats. The primary changes we made include the addition of some battery protection circuitry as well as the controls necessary to pilot the autonomous submarine used in the interoperability challenge.

#### A. Electrical Box

Many of the components that make up Thurman's electrical system are the same or similar to previous boat's electrical systems. The main difference between Thurman and her predecessors is the location of the electrical box within the hull. This year we created a custom box for our electrical system which can fit down inside the hull so the top of the electrical box is flush with the deck. This allows us additional deck space for the submarine and lowers our center of gravity.

#### B. Motherboard and Sensors

This year Thurman's electrical system was housed within an aluminum box (23" x 15" x 7"). This box was designed and constructed so as to be recessed within the hull of the boat.

Computationally, Thurman has an Intel Core i7 processor mounted on an ASUS Z97 Extreme6 motherboard and has 8 GBs of high-speed RAM. This formidable set-up allows for multiple processor intensive programs such as image processing to run at a high rate at the same time without any problems.

Thurman is equipped with a collection of sophisticated sensors that allow her to function autonomously. Thurman uses: a Hokuyo UTM-30LX LIDAR, a Point Grey Firefly 1.3PM USB camera, a Garmin 19x HVS GPS, a KVH DSP-3000 fiber optic gyroscope, a Aquarian Audio Products H2a high impedance hydrophone, and a Crenova iScope 2.0 MP Endoscope Snake Camera.

Thurman uses a GPS, and FOG in conjunction to identify her relative position and orientation in space. Additionally, she uses a single camera and LIDAR together to allow her to visually and spatially perceive her surroundings. By using each of these sensors, Thurman is able to understand and react to her surrounding environment.

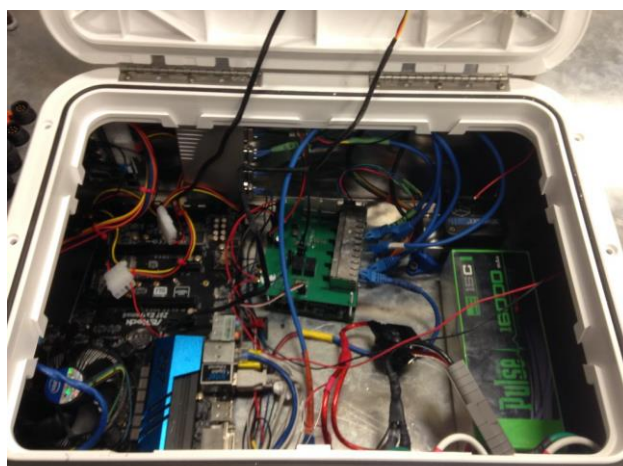


Fig. 7. The electrical box is recessed down inside the hull and can be accessed via a lid that is flush with the deck.

She uses the information from the GPS to determine her location and speed. The FOG allows her to detect instantaneous change in her orientation on the horizontal plane. Each of these sensors communicates with Thurman's computer via a serial RS 232 signal.

The LIDAR is rotated over a 0.2 radian arc by a Dynamixel AX-12 servo to obtain a 3D point cloud from the planar LIDAR sensor.

The hydrophone is used to locate the pinger. The endoscope snake camera is used by Thurman's submarine to transmit images back to her during the interoperability challenge.

#### C. Custom PCB

This year we updated the design for our custom pcb to include some additional functionality. This year our custom pcb acts as our main voltage bus, e-stops our boat once one of the estop buttons is pressed, and provides some protection circuitry for our battery. The voltage bus and e-stop circuitry are mostly the same as the previous iteration, but with additional fuses to prevent any of the lines on our voltage bus from drawing too much current.

The battery protection circuitry is new this year, and is designed to prevent the boat from draining the battery too low. This circuitry uses a tri-state buffer and a voltage divider to provide the current battery voltage as the input to an Arduino. Once the Arduino detects that the battery has dropped below a safe operating voltage threshold, it triggers the e-stop, causing the thrusters to lose power. Since the thrusters provide our

largest power draw, by turning them off once a threshold voltage is reached we are able to prevent the battery from being drained to an unsafe level.

#### D. Interoperability

As described in Section II, D we chose to use a Huiqi HQ2011 RC submarine for the interoperability challenge. Because this uses a proprietary communication protocol we are unable to use another board to communicate with it over the same communication protocol. Additionally, the sub is too small to fit another board inside it to replace the current electronics so that we could communicate with the sub. In the absence of these options, we chose to pilot the sub using the provided remote control. In order to actuate this remote control we chose to replace each of the buttons on the controller with a relay. Each of these relays is controlled by an Arduino, allowing us autonomously pilot the submarine.

In order to capture images of the underwater hex display we are using a Crenova iScope usb endoscope. This camera is waterproof and has a 5m usb cable. We have mounted this camera onto the roof of the submarine and it also acts as a tether which attaches the sub to the boat.

#### E. Networking

Much like last year we are using a pair of Ubiquiti Rocket M5's to establish communication between the boat and our base station on shore. The Rocket on shore is attached to a laptop via Ethernet. On the boat we use a power-over Ethernet adapter to power the Rocket from our battery. The Rockets operate in the 5470MHz-5825MHz frequency range and there is an antenna both on the boat and on the base station on shore to allow communication.

### IV. VISION AND PERCEPTION

Throughout the challenge course there are a variety of landmarks and obstacles that our boat must detect and correctly interact with. In order to accomplish this Thurman has a variety of different detectors used to observe her surroundings and identify features. These detectors will be described in the following sections.

#### A. Buoy Detection

Thurman detects buoys using a late fusion approach that combines the classifications from the camera and LIDAR systems. This allows the buoy detection system to utilize both the color and pattern information in the camera images while at the same time making use of the real world locations of the objects detected with the LIDAR.

In order to incorporate color information into the camera-based buoy detection, we shifted from calculating various local binary patterns (LBP) [1] in grayscale to automatically learning effective color features with a neural network. The neural network affords the flexibility of being able to learn arbitrary features given sufficient training data, as well as combining the steps of detecting buoys and determining their color.

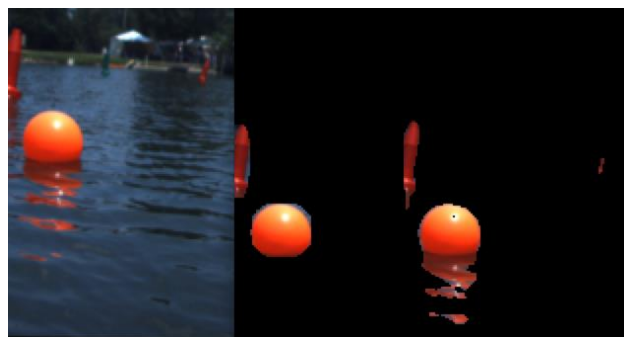


Fig. 8. The neural network recognizes the red buoys and bells. Left: the original image; middle: the ground truth labels; right: the network's predictions.

On the LIDAR side, Thurman uses a unionfind algorithm to cluster sufficiently close points. Several heuristics are then applied to filter for the clusters representing buoys. First, clusters with too many or too few points are discarded. Next the computed radius for each cluster is checked to lie within a given interval. The remaining clusters are brought to image coordinates by applying a transform to the cluster centers.

The fusion consists of matching cluster centers to the nearest buoy found in the camera image. Pairs of camera buoys and LIDAR clusters that are sufficiently close to Thurman are then considered buoys.

#### B. Buoy Tracking

Buoys are tracked using template matching with each buoy as template images and matching them with camera images.

The buoy tracking algorithm takes in (1) the current and next camera images and (2) a list of bounding boxes of buoys that were detected with the buoy detection algorithm on the current image.

The algorithm loops through these bounding boxes. For each bounding box, the buoy is cut out of the current image and used as a template image. The template image is matched against the next image using OpenCV's template matching algorithm. This algorithm basically matches the template image against all possible sub-images of the same size in the next image and scores each sub-image depending on how closely they match with a score of 0 (complete mismatch) to 1 (complete match). The sub-image with the highest score will be returned and used as the new bounding box for the buoy.

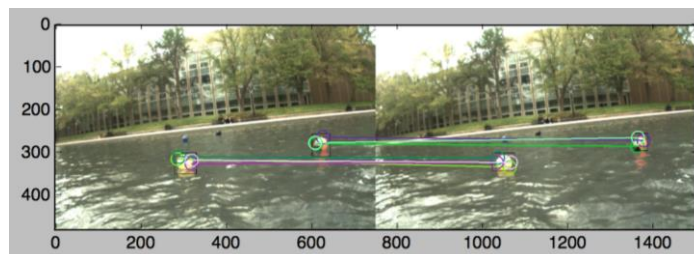


Fig. 9. The two images show a pair of buoys identified and tracked from one frame to the next.

To account for the buoy getting closer or further away, the algorithm also runs OpenCV's template matching algorithm on rescaled versions of the template images. The template images are rescaled by a few pixels only, since the buoy will not move far between camera images. The match with the highest template matching score among the rescaled versions



will be returned and used as the new bounding box for the buoy.

Running the algorithm against many template images is slow and inefficient. Thus, there is a threshold value for the template matching score. If the matching score passes this threshold value, that match will be used as the new bounding box. This threshold value is experimentally determined.

### C. Sign Detection

There is a two-step process for sign detection. Signs are first detected solely with the LIDAR. They are then matched to detected shapes (if any) in the image from the camera.

Like the buoy detection, sufficiently close LIDAR points are grouped together into segments using Union-Find. Then, using the x and y dimensions of those points, a line of best fit is found using Deming (orthogonal) regression. If the linear fit is good enough, the average z value (height) of the points in the segment is sufficiently high, and the largest distance between any two points (size of the sign) is sufficiently low, then the segment is considered a sign. The slope of the linear fit is also used to calculate the angle of the sign relative to Thurman, for use in the route planning.

The location of the sign is then transformed into pixel coordinates. For each shape detection, the closest unmatched sign is found. If they are sufficiently close, the sign is considered to contain that shape. The shape detections that don't match up are not considered.

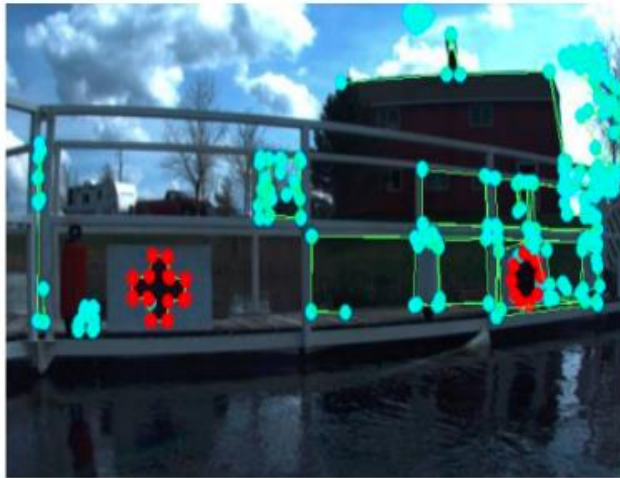


Fig. 10. This image shows the shapes identified on the signs found at the dock.

Thurman accomplishes shape detection using the OpenCV [2] library. OpenCV image processing filters such as the Gaussian blur filter and the bilateral filter (noise reduction) are used to help improve the edge detection results from the canny filter. Once the edges have been obtained, the OpenCV contour detection algorithm is used to determine significant contours from those edges. The approxPolyDP function is then used to approximate the corners to the best-fitting polygon for each contour. The corners are then used to form the edges of the unknown shape. These edges are cross-checked with the parameters of a triangle, cross, or circle to determine what shape it is. The contour images are further analyzed to determine which color group each shape belongs to.

### D. Gate Detection

To detect the gates, we consider all of the buoys that Thurman has seen so far in her run. First, all of the buoys whose radii are too small are discarded. Then, every subset consisting of one red buoy, two white buoys, and one green buoy is considered. Our goal is to find the subset(s) we are most confident that are red, white, white, green buoy gates. We represent this confidence as a real number between 0 and 1, with 1 being very confident, and 0 being not at all confident.

For each subset, we first find the maximum distance between any pair of buoys. If the maximum distance is too large, the subset is discarded. Otherwise, a line of best fit is found on the x and y coordinates of the buoys using Deming (orthogonal) regression. The confidence for this subset is set to the "R squared" value for the linear fit. Then, we check if the buoys are in the correct color order along the line. If they aren't, they are discarded. Finally, to check how even the spacing is between the buoys, we find the distances between each pair of adjacent buoys. We multiply the confidence by the minimum pair distance divided by the maximum pair distance. If a subset makes it to the end of that without being discarded, it is considered a gate with the corresponding confidence.

### E. Interoperability Challenge

In order to complete the Hex character identification portion, a small waterproof camera is mounted on the top of the UAV pointed forward. The UAV scans the area and takes several pictures. These pictures are used to determine where the LEDs are and to maneuver the sub so as to position the LED's at the center of the image, ensuring that all LED's are seen.

After the LED's are at the center of the image, the camera takes the picture that will be used to determine the displayed character. This image is fed into an OCR program and the results are sent to the server for verification.

## V. SLAM

Thurman utilizes an Incremental Smoothing and Mapping (iSAM) approach to solving the SLAM problem. Due to the relative sparsity of obstacles, we can easily define features and setup the graph structure that iSAM requires to operate. Additionally, the sparsity of features allows us to prevent overconstraining the graph, decreasing the computational power needed to smooth and optimize the map.

Rather than use our own backend for processing and manipulating the graph, we use the same backend used by the Perceptual Robotics Lab (PeRL) run by Ryan Eustice at the University of Michigan, which we interact with over LCM. This backend allows us to focus on the feature detection and graph structure. By using a verified backend we have been able to simplify debugging and implementing this version of SLAM.

For solving the data correspondence problem, a simple algorithm that compares the location at which the buoy is detected to the location of previously seen buoys of the same color. If they are within a certain distance, the detections are combined and the graph is further constrained.

## VI. CHALLENGE STATIONS

In order to address the changes to the challenge stations this year, the way we approach each challenge station has changed. The main changes will be described in the following section.

### A. Start/Speed Gates

Thurman listens to the buoy detections to determine which buoys are detected as speed gates. Filtering only the speed gate buoy detections, she proceeds forward based on what was detected. If both a red and a green were detected to form a pair/gate, Thurman heads straight between the two. If only a red or green buoy was detected, Thurman heads slightly to the right or left of the single buoy respectively. In the worst case, if no buoys were detected, Thurman heads straight in the direction it was deployed.

### B. Obstacle Course

For this task Thurman listens to detections of buoys in the Red-White-White-Green (RWWG) pattern and from this heads toward the gate specified in the JSON message. While avoiding obstacles, Thurman maximizes her available range of motion by keeping near to the calculated center of the channel, and at the same time examining sets of buoys matching the RWWG pattern. The channel is exited just as it is entered.

### C. Docking

The docking task begins by navigating to the area near the dock based on the given GPS coordinates. Camera and LIDAR are then used to identify possible signs on the dock via the shape detection algorithm. The shape and color of each sign detected is used along with an orthogonal regression to determine which of the detected signs make up the dock. Two signs with a unique shape and color are chosen as the docking points, based on the desired docking locations obtained from the competition server. Thurman then drives to the first of these signs and drives along the dock for a bit to attach the Velcro to the strip on the dock. Finally, the boat backs up to its original approach position before driving to the base of the second sign and repeating the same action to deposit a second strip of Velcro.

### D. Acoustic Beacon

Thurman uses a modified breadth-first search algorithm to find the acoustic beacons. She visits each buoy and uses the hydrophone to take a reading of the sound intensity. From there the boat will navigate to the next unmeasured buoy (backtracking if necessary) and continue to take readings until all buoys are found. Once all are located, the boat picks the buoy associated with the highest intensity reading for each of the given frequencies and travels in a circle around each one to signal the beacons have been found. While this isn't the most time efficient solution that we worked on, it had the highest success rate during simulation runs.

## VII. UTILITIES

### A. Bot-Procman

Due to the design, our process manager can be used both for test/debug sessions and competition runs. The process manager is invoked with a configuration file as an argument.

The configuration file names all the processes that can be run. Because the configuration is not hard-coded, we can use a different configuration when testing as opposed to a trial run, affording us great flexibility during both runtime and testing sessions. In addition, the process manager sports an interactive graphical user interface for managing the processes. This allows the developer to start and stop a process or view the process's output without searching for the terminal it was started from. This aids us in debugging any problems Thurman may encounter because we can see the sensor data as well as any LCM messages being sent.

The process manager also attempts to ensure that processes are running and working as intended. For this, the process manager will restart a process if it crashes for whatever reason. Also, the process manager listens to the LCM channels of the managed processes. Using the frequency of publishes the process manager can determine if a process has become stuck even if it has not terminated. These features aim to improve robustness by preventing a total system failure due to a minor bug.

### B. Vis

There are too many LCM messages at one time to quickly understand as a human. To quickly gain knowledge of what the boat is doing at a certain time, we have developed a visualization system. In this environment, we can see what is within SLAM. This includes a 3D environment, where we can pan and zoom onto locations of the boat, buoys, and other challenge station elements. This aids the debugging process, since we can quickly see what led the boat to performing unwanted behavior.

## VIII. CONCLUSION

Thurman underwent some major changes from her predecessors including a new hull shape and electrical box placement, as well as some new detection algorithms and SLAM software. The new hull design lowers Thurman's center of gravity as well as allowing the electrical box to be recessed into the hull. These changes help Thurman to be more successful than her predecessors when completing challenge stations and navigating the competition course.

## IX. REFERENCES

- [1] S. Liao, X. Zhu, Z. Lei, L. Zhang and S. Z. LI, "Learning multi-scale block local binary patterns for facial recognition.," *Advances in Biometrics*, pp. 828-837, 2007.
- [2] OpenCV [computer software library], 2000, Intel Corporation, Willow Garage, Itseez.